Statistics Toolbox™ Release Notes

How to Contact MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support
`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

☎ 508-647-7000 (Phone)

508-647-7001 (Fax)

✉ The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Statistics Toolbox™ Release Notes

Trademarks

Patents

Contents

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

R2009a

R2008b

R2008a

R2007b

R2007a

R2006b

R2006a

R14SP3

R14SP2

R2014a

Version: 9.0

New Features: Yes

Bug Fixes: Yes

Repeated measures modeling for data with multiple measurements per subject

`fitrm` is a new function for fitting models to repeated measures data, where each subject has multiple response measurements. It produces an object of the new `RepeatedMeasuresModel` class. You can:

- Perform analysis of variance for between-subjects factors using `anova`.
- Perform multivariate analysis of variance using `manova`.
- Perform hypothesis tests on the coefficients using `coeftest`.
- Perform repeated measures analysis of variance using `ranova`.
- Test for sphericity (compound symmetry) with Mauchly's test using `mauchly`.
- Plot data and estimated marginal means with optional grouping using `plot` and `plotprofile`.
- Compute summary statistics organized by group using `grpstats`.
- Perform multiple comparisons of marginal means using `multcompare`.
- Make predictions on new data with the fitted repeated measures model using `predict`.
- Generate random data with the fitted repeated measures model at new design points using `random`.

For the properties and methods of this object, see the `RepeatedMeasuresModel` class page.

`fitcsvm` function for enhanced performance of support vector machines (SVMs) for binary classification

You can now use the new `fitcsvm` function to train an SVM classifier for one- or two-class learning. `fitcsvm` creates an object of the new class `ClassificationSVM` or existing class `ClassificationPartitionedModel`.

ClassificationSVM is a new class for accessing and performing operations on the training data. CompactClassificationSVM is a new class for storing configurations of trained models without storing training data. The syntax and methods resemble those in the existing ClassificationTree and CompactClassificationTree classes.

The new fitcsvm function and ClassificationSVM and CompactClassificationSVM classes include the functionality of the svmtrain and svmclassify functions. ClassificationSVM provides several benefits compared to the svmtrain and svmclassify functions:

- The new functionality
  - Supports computation of soft classification scores
  - Supports fitting posterior probabilities
  - Has improved training speed, especially on big data with well-separated classes by providing shrinkage
  - Allows a warm restart by accepting an initial α value
  - Allows training to resume after the maximum number of iterations is exceeded
  - Supports robust learning in the presence of outliers
- ClassificationSVM is built on the same framework as ClassificationTree, ClassificationDiscriminant, and ClassificationKNN, so you have a variety of options and methods, including:
  - Cross validation
  - Resubstitution statistics
  - Generalization statistics
  - Weighted classification

For all methods and properties of the new objects, see the ClassificationSVM and CompactClassificationSVM class pages.

`evalclusters` methods to expand the number of clusters and number of gap criterion simulations

Compatibility Considerations: Yes

There are two new methods for the objects created using the `evalclusters` function:

- `addK` adds additional number of clusters to be evaluated. This method applies to all classes of cluster evaluation (i.e., `clustering.evaluation.GapEvaluation`, `clustering.evaluation.SilhouetteEvaluation`, `clustering.evaluation.CalinskiHarabaszEvaluation`, and `clustering.evaluation.DaviesBouldinEvaluation`).
- `increaseB` increases the number of reference data sets for gap criterion simulations. This method applies to the `clustering.evaluation.GapEvaluation` class.

The default value of the `'SearchMethod'` name-value pair argument for `clustering.evaluation.GapEvaluation` objects is now always `'globalMaxSE'`.

Compatibility Considerations

The default value of the `'SearchMethod'` name-value pair argument for `clustering.evaluation.GapEvaluation` objects is now always `'globalMaxSE'` and does not change depending on the value of the `'KList'` name-value pair argument.

p-value output from the `multcompare` function

Compatibility Considerations: Yes

`multcompare` now returns the *p*-value of each pairwise comparison of group means. `multcompare` returns the *p*-value in the sixth column of its first output argument. The *p*-value is the overall significance level at which the individual comparison is borderline significant.

Compatibility Considerations

The first output argument of `multcompare` now has six columns, instead of five. The sixth column contains the *p*-value.

`mnrfit`, `lassoglm`, and `fitglm` functions accept categorical variables as responses

`mnrfit` now accepts a categorical variable as the response. The `lassoglm`, `fitglm`, and `glmfit` functions now accept a two-level categorical variable as the response. The `random` method for the `GeneralizedLinearModel` class now also returns categorical responses.

Functions accept `table` inputs as an alternative to dataset array inputs

The following functions and methods now accept table inputs as alternative to dataset array inputs.

Functions and Methods	Class
`fitlm`, `fitglm`, `fitlme`, `fitnlm`, `stepwiseglm`, `stepwiselm`, `grpstats`, `datasample`	N/A
`predict`, `random`, `feval`	`LinearModel`
`devianceTest`, `random`, `predict`, `feval`	`GeneralizedLinearModel`
`random`, `predict`, `feval`	`NonLinearModel`
`random`, `predict`	`LinearMixedModel`

Functions and model properties return a `table` rather than a dataset array

Compatibility Considerations: Yes

The following functions, methods, and model properties now return a table rather than a dataset array.

Functions and Methods	Class
xptread, grpstats*	N/A
anova	LinearModel
devianceTest	GeneralizedLinearModel
fixedEffects, randomEffects	LinearMixedModel

Property	Class
VariableInfo, ObservationInfo, Variables, Diagnostics, Residuals, Coefficients	LinearModel
VariableInfo, ObservationInfo, Variables, Diagnostics, Residuals, Fitted, Coefficients	GeneralizedLinearModel
VariableInfo, ObservationInfo, Variables, Diagnostics, Residuals, Coefficients	NonLinearModel
VariableInfo, ObservationInfo, Variables, Coefficients, ModelCriterion	LinearMixedModel

*grpstats now matches the output with input type.

Compatibility Considerations

The functions and properties listed now return a table instead of a dataset array. You can convert them to dataset arrays using the table2dataset function.

Default value of 'EmptyAction' on kmeans is now 'singleton'.

Compatibility Considerations: Yes

The default value of the 'EmptyAction' name-value pair argument of the kmeans function is now 'singleton'.

Compatibility Considerations

To set the value of 'EmptyAction' to 'error', you must explicitly specify 'EmptyAction','error'.

Functions for classification methods and clustering

The following are new functions for classification and regression trees, discriminant analysis, nearest neighbors, Naive Bayes classification, and Gaussian mixture models.

New Function	Replacing
fitcdiscr	ClassificationDiscriminant.fit
fitcknn	ClassificationKNN.fit
fitctree	ClassificationTree.fit
fitrtree	RegressionTree.fit
fitNaiveBayes	NaiveBayes.fit
fitgmdist	gmdistribution.fit
templateDiscriminant	ClassificationDiscriminant.template
templateKNN	ClassificationKNN.template
templateTree	ClassificationTree.template or RegressionTree.template
makecdiscr	ClassificationDiscriminant.make

Functionality being changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
ClassificationDiscriminant.fit	Still runs	fitcdiscr	Replace instances of ClassificationDiscriminant.fit with fitcdiscr.
ClassificationKNN.fit	Still runs	fitcknn	Replace instances of ClassificationKNN.fit with fitcknn.
ClassificationTree.fit	Still runs	fitctree	Replace instances of ClassificationTree.fit with fitctree.
RegressionTree.fit	Still runs	fitrtree	Replace instances of RegressionTree.fit with fitrtree.
NaiveBayes.fit	Still runs	fitNaiveBayes	Replace instances of NaiveBayes.fit with fitNaiveBayes.
gmdistribution.fit	Still runs	fitgmdist	Replace instances of gmdistribution.fit with fitgmdist.
ClassificationDiscriminant.template	Still runs	templateDiscriminant	Replace instances of ClassificationDiscriminant.tem with templateDiscriminant.
ClassificationKNN.template	Still runs	templateKNN	Replace instances of ClassificationKNN.template with templateKNN.

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
ClassificationTree.template or RegressionTree.template	Still runs	templateTree	Replace instances of ClassificationTree.template or RegressionTree.template with templateTree.
ClassificationDiscriminant.make	Still runs	makecdiscr	Replace instances of ClassificationDiscriminant.mak with makecdiscr.

R2013b

Version: 8.3

New Features: Yes

Bug Fixes: Yes

Linear mixed-effects models

`LinearMixedModel` is a new class for fitting linear mixed-effects (LME) models. Fit multi-level LME models or LME models with nested and/or crossed random effects using the `fitlme` or `fitlmematrix` function. You can:

- Specify LME models using either the formula notation or via matrix input.
- Fit LME models using maximum likelihood (ML) or restricted maximum likelihood (REML).
- Specify a covariance pattern for the random effects.
- Calculate estimates of best linear unbiased predictors (BLUPs) for random effects.
- Perform custom joint hypothesis tests on fixed and random effects.
- Compute confidence intervals on fixed effects, random effects, and covariance parameters.
- Examine residuals, diagnostic plots, fitted values, and design matrices.
- Compare two different models via theoretical or simulated likelihood ratio tests.
- Make predictions on new data using the fitted LME model.
- Generate random data using the fitted LME model at new design points.

For the properties and methods of this object, see the class page for `LinearMixedModel`.

Code generation for probability distribution and descriptive statistics functions (using MATLAB Coder)

Many probability distribution and descriptive statistics functions are now supported for code generation. For a full list of Statistics Toolbox functions that are supported by MATLAB® Coder™, see Statistics Toolbox Functions.

`evalclusters` function for estimating the optimal number of clusters in data

The new function `evalclusters` estimates the optimal number of clusters for various criterion values, and returns the clustering solution corresponding to the estimated optimal value.

You can provide clustering solutions, ask `evalclusters` to use one of the built-in clustering algorithms, `'kmeans'`, `'linkage'`, or `'gmdistribution'`, or provide a function handle.

The following criteria are available:

- The Calinski-Harabasz (CH) index
- The Silhouette index
- The Gap statistic
- The Davies-Bouldin (DB) index

`mvregress` function that now accepts a design matrix even if `Y` has multiple columns

`mvregress` now accepts an n-by-$(p + 1)$ design matrix `X`, when the response `Y` is an n-by-d matrix with $d > 1$, where n is the number of observations, p is the number of predictor variables, d is the number of dimensions in the response, and `X` includes a column of ones for the intercept (constant) term.

Upper tail probability calculations for cumulative distribution functions

Statistics Toolbox now provides upper tail probability calculations for cumulative distribution functions. You can compute the upper tail probabilities using a trailing `'upper'` argument in the following functions:

- `cdf` function for probability distribution objects, returned by `pd = makedist(distname)` or `pd = fitdist(X,distname)`:

  ```
  cdf(pd,X,'upper')
  ```

- `cdf` function:

  ```
  Y = cdf('name',X,A,'upper')
  ```

  ```
  Y = cdf('name',X,A,B,'upper')
  ```

  ```
  Y = cdf('name',X,A,B,C,'upper')
  ```

- Distribution-specific `cdf` functions:

Distribution	**New Syntax**
Beta	`p = betacdf(X,A,B,'upper')`
Binomial	`Y = binocdf(X,N,P,'upper')`
Chi-square	`p = chi2cdf(X,V,'upper')`

Extreme Value	`P = evcdf(X,mu,sigma,'upper')`
	`[P,PLO,PUP] = evcdf(X,mu,sigma,pcov,'upper')`

Exponential	`P = expcdf(X,mu,'upper')`
	`[P,PLO,PUP] = expcdf(X,mu,pcov,'upper')`

F	`P = fcdf(X,V1,V2,'upper')`

Gamma	`P = gamcdf(X,A,B,'upper')`
	`[P,PLO,PUP] = gamcdf(X,A,B,pcov,'upper')`

Geometric	`Y = geocdf(X,P,'upper')`
Generalized Extreme Value	`P = gevcdf(X,k,sigma,mu,'upper')`
Generalized Pareto	`P = gpcdf(X,sigma,theta,'upper')`
Hypergeometric	`P = hygecdf(X,M,K,N,'upper')`

Lognormal	`P = logncdf(X,mu,sigma,'upper')`
	`[P,PLO,PUP] = logncdf(X,mu,sigma,pcov,'upper')`

Negative Binomial	`Y = nbincdf(X,R,P,'upper')`
Non-central F	`P = ncfcdf(X,NU1,NU2,DELTA,'upper')`

Distribution	**New Syntax**
Non-central t	`P = nctcdf(X,NU,DELTA,'upper')`
Non-central Chi-square	`P = ncx2cdf(X,V,DELTA,'upper')`

Normal	`P = normcdf(X,mu,sigma,'upper')`
	`[P,PLO,PUP] = normcdf(X,mu,sigma,pcov,'upper')`

Poisson	`P = poisscdf(X,lambda,'upper')`
t	`P = tcdf(X,V,'upper')`
Rayleigh	`P = raylcdf(X,B,'upper')`
Uniform Discrete	`P = unidcdf(X,N,'upper')`
Uniform Continuous	`P = unidcdf(X,A,B,'upper')`

Weibull	`P = wblcdf(X,A,B,'upper')`
	`[P,PLO,PUP] = wblcdf(X,A,B,pcov,'upper')`

`partialcorri` function for partial correlation with asymmetric treatment of inputs and outputs

The new function `partialcorri` computes linear partial correlation coefficients with internal adjustments. You can compute partial correlation between pairs of variables in `Y` and `X`, adjusting for the remaining variables in `X`, or between pairs of variables in `Y` and `X`, adjusting for the remaining variables in `X`, after first controlling both `X` and `Y` for the variables in `Z`.

You can also:

- Specify whether to use Pearson or Spearman partial correlations.
- Specify how to handle missing values.
- Perform hypotheses test of zero correlation against a one-sided or two-sided alternative.

Fitting functions for linear, generalized linear, and nonlinear models

There are new functions for the fitting and stepwise algorithms of linear and generalized linear models, and the fitting algorithm of nonlinear models. The new functions are as follows.

New Function	Replacing
`fitlm`	`LinearModel.fit`
`stepwiselm`	`LinearModel.stepwise`
`fitglm`	`GeneralizedLinearModel.fit`
`stepwiseglm`	`GeneralizedLinearModel.stepwise`
`fitnlm`	`NonLinearModel.fit`

Functionality being changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
`LinearModel.fit`	Still runs	`fitlm`	Replace instances of `LinearModel.fit` with `fitlm`
`LinearModel.stepwise`	Still runs	`stepwiselm`	Replace instances of `LinearModel.stepwise` with `stepwiselm`
`GeneralizedLinearModel.fit`	Still runs	`fitglm`	Replace instances of `GeneralizedLinearModel.fi` with `fitglm`

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
GeneralizedLinearModel.stepwise	Still runs	stepwiseglm	Replace instances of GeneralizedLinearModel.ste with stepwiseglm
NonLinearModel.fit	Still runs	fitnlm	Replace instances of NonLinearModel.fit with fitnlm

R2013a

Version: 8.2

New Features: Yes

Bug Fixes: Yes

Support vector machines (SVMs) for binary classification (formerly in Bioinformatics Toolbox)

Support vector machines are now in Statistics Toolbox™. Train support vector machine classifier using `svmtrain` and classify data using `svmclassify`.

Probabilistic PCA and alternating least-squares algorithms for principal component analysis with missing data

Two new features handle missing data in principal component analysis:

- The new function `ppca` uses probabilistic principal components analysis, which is based on an isotropic error model.
- The function `pca` has a new alternating least squares (ALS) algorithm. Use the name-value pair argument `'algorithm'` with the value `'als'`.

Anderson-Darling goodness-of-fit test

The new function `adtest` performs the Anderson-Darling goodness-of-fit test. `adtest` can perform:

- Simple test: Test against a specific distribution with parameters specified. You can test against any continuous univariate parametric distribution.
- Composite test: Test against a specified distribution family (also called an omnibus test). You can test against the normal, exponential, extreme-value, lognormal, or weibull distribution families.

Decision-tree performance improvements and categorical predictors with many levels

- The training speed for decision trees and their ensembles is improved. The improvement is best seen in decision tree ensembles obtained using the `fitensemble` function or `TreeBagger` class.
- Improved efficiency of `TreeBagger` when used in parallel mode.

- You can specify the number of surrogate splits saved in decision trees using the 'surrogate' name-value pair argument in the fit and template methods of the ClassificationTree and RegressionTree classes.
- ClassificationTree.fit and ClassificationTree.template provide several heuristic methods for splitting on categorical predictors with many levels. Use the 'AlgorithmForCategorical' name-value pair argument to specify the algorithm to find the best split and the 'MaxCat' name-value pair argument to specify the maximum number of categories you allow.

Grouping and kernel density options in scatterhist function

The scatterhist function has these name-value pair arguments:

- 'Group' lets you specify a grouping variable and produces a grouped scatter plot.
- 'Kernel' lets you use grouped kernel density plots instead of overall histograms for the marginal distributions.
- Additional options let you change colors, line properties, legends, and more.

Nonlinear model enhancements

These functions now accept additional error models and fixed or fit-dependent weights.

NonLinearModel methods: • NonLinearModel.fit • predict • random	• Use the 'ErrorModel' and 'ErrorParameters' name-value pair arguments to define the error models and 'Weights' to enter weights. • The NonLinearModel object has a new property, WeightedResiduals.
nlinfit	• Use 'ErrorModel' and 'ErrorParameters' name-value pair arguments to define the error models and 'Weights' name-value pair to enter weights.

	• `nlinfit` returns a structure containing information about the error model you define.
`nlpredci`	• Accepts the error model structure returned by `nlinfit`. • Adjusts Scheffe type simultaneous confidence intervals for weights, error models, and rank deficient Jacobians.

Additional functionality changes are:

- `disp` (`NonLinearModel` method) shows only estimable coefficients, and shows `NaN` for inestimable coefficients.
- Ftest (`NonLinearModel` method) automatically decides whether to compare the full model against an intercept-only model or zero.
- `NonLinearModel` properties such as `Diagnostics`, `Residuals`, `LogLikelihood`, `SSE`, and `SST` account for weights and error models.

Syntax changes in parametric hypothesis test functions

Parametric hypothesis test functions accept optional input arguments as name-value pair arguments.

`adtest`	Anderson-Darling goodness-of-fit test
`ansaribradley`	Ansari-Bradley test
`dwtest`	Durbin-Watson test
`kstest`	One-sample Kolmogorov-Smirnov test
`kstest2`	Two-sample Kolmogorov-Smirnov test
`lillietest`	Lilliefors test
`ttest`	One-sample *t*-test
`ttest2`	Two-sample *t*-test
`vartest`	One-sample variance chi-square test

`vartest2`	Two-sample variance *F*-test
`vartestn`	Variance test across multiple groups
`ztest`	*z*-test

Probability distribution enhancements

Compatibility Considerations: Yes

New probability distribution objects provide the following new functionality:

- Create a distribution without fitting to data using the new `makedist` function.
- Assign directly to parameter values.
- Create truncated distributions.
- Create and operate on arrays of distribution objects.
- Create custom distributions. To begin, use `dfittool` and select **Edit > Define Custom Distributions**. Use the provided template to define the `'Laplace'` distribution, or modify it to create your own.
- Compute and plot likelihood ratio confidence intervals and profile likelihood for fitted probability distributions.
- Additional distributions in the probability distribution framework:
  - Multinomial
  - Piecewise Linear
  - Triangular
  - Uniform

You can continue fitting distributions to data using the existing `fitdist` function.

Compatibility Considerations

The class names of probability distribution objects returned by `fitdist` are different than in earlier releases.

R2012b

Version: 8.1

New Features: Yes

Bug Fixes: Yes

Boosting algorithms for imbalanced data, sparse ensembles, and multiclass boosting, with self termination

There are three new boosting algorithms for classification:

- **RUSBoost** (boosting by random undersampling) for imbalanced data (data in which one class has many more observations than the other).
- **LPBoost** (linear programming) and **TotalBoost** (totally corrective boosting) which self-terminate, can lead to a sparse ensemble, and can be used for multiclass boosting.

Burr distribution for expressing a wide range of distribution shapes while preserving a single functional form for the density

There is a new probability distribution object for the Burr Type XII distribution, a three-parameter family of continuous distributions on the real line. Use `fitdist` to fit this distribution to data. Use `ProbDistUnivParam` to specify the distribution parameters directly. Either function produces a distribution you can use to generate random samples or compute functions such as `pdf` and `cdf`.

Data import to a dataset array with the MATLAB Import Tool

You can now import data from a file directly into a `dataset` array using the MATLAB Import Tool.

Principal component analysis enhancements for handling NaN as missing data, weighted PCA, and choosing between EIG or SVD as the underlying algorithm

Compatibility Considerations: Yes

The new pca function includes additional functionality for principal component analysis. Features of pca include:

- Handling of NaN as missing data values.
- Weighted principal component analysis with user-specified weights.
- Choice of SVD or EIG algorithm for computing principal components.
- Option to specify number of components to return.
- Option to not center before computing principal components.

Compatibility Considerations

The new pca function replaces the princomp function.

Speedup of k-means clustering using Parallel Computing Toolbox

Statistics Toolbox now supports parallel execution for kmeans.

One-sided nonparametric hypothesis tests

An option to test one-sided right- or left-tailed alternatives is available for these nonparametric hypothesis tests:

- signrank
- ranksum
- signtest

Reorder nodes in dendrogram plots

- The dendrogram function has new options for reordering the nodes of hierarchical binary cluster trees:

  - The reorder option allows you to specify a permutation vector for the order of nodes in a dendrogram plot.
  - The checkcrossings option checks whether a requested permutation vector leads to crossing branches in a dendrogram plot.
- The function optimalleaforder generates an optimal permutation of nodes.

Nonlinear model enhancements

Compatibility Considerations: Yes

You can add a vector of observation weights, or a handle to a function that returns a vector of observation weights, to these functions:

- NonLinearModel.fit.
- predict and random (NonLinearModel methods).
- nlinfit and nlpredci.

For an example of weighted fitting, see Weighted Nonlinear Regression.

Compatibility Considerations

Use either Weights or RobustWgtFun when performing weighted nonlinear regression.

Changes to LinearModel diagnostics

Compatibility Considerations: Yes

The diagnostics in the Diagnostics dataset array for LinearModel objects are in a new order, and no longer appear in the Variables editor. The new order is:

- Leverage
- CooksDistance
- Dffits
- S2_i
- CovRatio

- `Dfbetas`
- `HatMatrix`

Compatibility Considerations

To access the correct diagnostics, you should update any code that indexes the diagnostics dataset array columns by number.

Functionality being changed

Compatibility Considerations: Yes

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
`princomp`	Still runs	`pca`	Replace instances of `princomp` with `pca`

R2012a

Version: 8.0

New Features: Yes

Bug Fixes: Yes

Linear, Generalized Linear, and Nonlinear Models for Regression

`LinearModel` is a new class for performing linear regression. `LinearModel.fit` creates a model that:

- Lets you fit models with both categorical and continuous predictor variables
- Contains information about the quality of the fit, such as residuals and ANOVA tables
- Lets you easily plot the fit
- Allows for automatic or manual exclusion of unimportant variables
- Enables robust fitting for reduced influence of outliers
- Lets you specify quadratic and other models using a symbolic formula
- Enables stepwise model selection

There are similar improvements for generalized linear and nonlinear modeling using the `GeneralizedLinearModel` and `NonLinearModel` classes. For details, see the class reference pages in the reference material, or Linear Regression, Stepwise Regression, Robust Regression — Reduce Outlier Effects, Generalized Linear Regression, or Nonlinear Regression in the User's Guide.

Variable Editor for Dataset Arrays

You can now edit, sort, plot, and select portions of dataset arrays from the MATLAB Variable Editor. For details, see Using Dataset Arrays in the User's Guide.

Lasso for Generalized Linear Regression

The `lassoglm` function regularizes generalized linear models. Use `lassoglm` to examine model alternatives and to constrain or remove redundant or unimportant variables in generalized linear regression. For details, see the function reference page, or Lasso Regularization of Generalized Linear Models in the User's Guide.

K-Nearest Neighbor Classification

`ClassificationKNN.fit` creates a classification model that performs *k*-nearest neighbor classification. You can check the quality of the model with cross validation or resubstitution. For details, see the `ClassificationKNN` page in the reference material, or Classification Using Nearest Neighbors in the User's Guide.

Random Subspace Ensembles

`fitensemble` can construct random subspace ensembles to improve the classification accuracy of both *k*-nearest neighbor classifiers and discriminant analysis classifiers. For details, see Ensemble Methods or Random Subspace Classification in the User's Guide.

Regularized Discriminant Analysis with Variable Selection

`ClassificationDiscriminant` models now have two parameters, `Gamma` and `Delta`, for regularization and lowering the number of variables. Set `Gamma` to regularize the discriminant. Set `Delta` to eliminate variables. Use `cvshrink` to obtain optimal `Gamma` and `Delta` parameters by cross validation. For details, see the reference pages, or Regularize a Discriminant Analysis Classifier in the User's Guide.

stepwisefit Coefficient History

The `stepwisefit` function now returns the fitted coefficient history in the `history.B` field.

RobustWgtFun Replaces WgtFun

Compatibility Considerations: Yes

The `WgtFun` option is now called `RobustWgtFun` in the `nlinfit`, `statget`, and `statset` functions. `RobustWgtFun` also makes the `Robust` option superfluous.

Compatibility Considerations

The WgtFun and Robust options are currently accepted by all functions. To avoid potential future incompatibilities, update code that uses the WgtFun and Robust options to use the RobustWgtFun option.

ClassificationTree Now Predicts Class with Minimal Misclassification Cost

Compatibility Considerations: Yes

The ClassificationTree predict method now chooses the class with minimal expected misclassification cost. Previously, it chose the class with maximal posterior probability. The new behavior is consistent with the cvLoss method. Furthermore, both ClassificationDiscriminant and ClassificationKNN predict using minimal expected misclassification cost. For details, see predict and loss.

Compatibility Considerations

If you use a nondefault cost matrix, some ClassificationTree classification predictions can differ from those in previous versions.

fpdf Improvements

The fpdf function now accepts a wider range of parameter values, including Inf.

R2011b

Version: 7.6

New Features: Yes

Bug Fixes: Yes

Lasso Regularization for Linear Regression

The `lasso` function incorporates both the lasso regularization algorithm and the elastic net regularization algorithm. Use `lasso` to remove redundant or unimportant variables in linear regression. The `lassoPlot` function helps you visualize `lasso` results, with a variety of coefficient trace plots and a cross-validation plot.

For details, see Lasso and Elastic Net.

Discriminant Analysis Classification Object

You can now use the `ClassificationDiscriminant` and `CompactClassificationDiscriminant` classes for classification via discriminant analysis. The syntax and methods resemble those in the existing `ClassificationTree` and `CompactClassificationTree` classes. The `ClassificationDiscriminant` class includes the functionality of the `classify` function. `ClassificationDiscriminant` provides several benefits compared to the `classify` function:

- After you fit a classifier, you can predict without refitting.
- `ClassificationDiscriminant` is built on the same framework as `ClassificationTree`, so you have a variety of options and methods, including:
  - Cross validation
  - Resubstitution statistics
  - A choice of cost functions
  - Weighted classification
- `ClassificationDiscriminant` can fit several models, including linear, quadratic, and linear or quadratic with pseudoinverse.

For details, see Discriminant Analysis.

Nearest Neighbor Searching for Points Within a Fixed Distance

The rangesearch function finds all members of a data set that are within a specified distance of members of another data set. As with the knnsearch function, you can set a variety of distance metrics, or program your own. rangesearch has counterparts that are methods of the ExhaustiveSearcher and KDTreeSearcher classes.

datasample Function for Random Sampling

The datasample function samples with or without replacement from a data set. It can also perform weighted sampling, with or without replacement.

Fractional Factorial Design Improvements

The fracfactgen function now allows up to 52 factors, instead of the previous limit of 26 factors. Specify factors as case-sensitive strings, using 'a' through 'z' for the first 26 factors, and 'A' through 'Z' for the remaining factors.

fracfact now checks for an arbitrary level of interaction in confounding, instead of the previous limit of confounding up to products of two factors. Set the MaxInt name-value pair to the level of interaction you want. You can also set names for the factors using the FactorNames name-value pair.

nlmefit Returns the Covariance Matrix of Estimated Coefficients

The nlmefit function now returns the covariance matrix of the estimated coefficients as the covb field of the stats structure.

signrank Change

The signrank test now defines ties to be entries that differ by 2*eps or less. Previously, ties were entries that were identical to machine precision.

Conversion of Error and Warning Message Identifiers

Compatibility Considerations: Yes

For R2011b, error and warning message identifiers have changed in Statistics Toolbox.

Compatibility Considerations

If you have scripts or functions that use message identifiers that changed, you must update the code to use the new identifiers. Typically, message identifiers are used to turn off specific warning messages, or in code that uses a `try/catch` statement and performs an action based on a specific error identifier.

For example, if you use the `'resubstitution'` method, the `'stats:plsregress:InvalidMCReps'` identifier has changed to `'stats:plsregress:InvalidResubMCReps'`. If you use the `'resubstitution'` method and your code checks for `'stats:plsregress:InvalidMCReps'`, you must update it to check for `'stats:plsregress:InvalidResubMCReps'` instead.

To determine the identifier for a warning, run the following command just after you see the warning:

```
[MSG,MSGID] = lastwarn;
```

This command saves the message identifier to the variable `MSGID`.

To determine the identifier for an error, run the following command just after you see the error:

```
exception = MException.last;
MSGID = exception.identifier;
```

Tip Warning messages indicate a potential issue with your code. While you can turn off a warning, a suggested alternative is to change your code so it runs warning free.

R2011a

Version: 7.5

New Features: Yes

Bug Fixes: Yes

Boosted Decision Trees for Classification and Regression

The new fitensemble function constructs ensembles of decision trees. It provides:

- Several popular boosting algorithms (AdaBoostM1, AdaBoostM2, GentleBoost, LogitBoost, and RobustBoost) for classification
- Least-squares boosting (LSBoost) for regression
- Most TreeBagger functionality for ensembles of bagged decision trees

There is also an improved interface for classification trees (ClassificationTree) and regression trees (RegressionTree), encompassing the functionality of classregtree.

For details, see Ensemble Methods.

Memory and Performance Improvements in Linkage Methods

The linkage and clusterdata functions have a new savememory option that can use less memory than before. With savememory set to 'on', the functions do not build a pairwise distance matrix, so use less memory and, depending on problem size, can use less time. You can use the savememory option when:

- The linkage method is 'ward', 'centroid', or 'median'
- The linkage distance metric is 'euclidean' (default)

For details, see the linkage and clusterdata function reference pages.

Conditional Weighted Residuals and Derivative Step Control in nlmefit and nlmefitsa

The nlmefit and nlmefitsa functions now provide the conditional weighted residuals of the fit. Use this information to assess the quality of the model; see Example: Examining Residuals for Model Verification.

The `statset` `Options` structure now includes `'DerivStep'`, which enables you to set finite differences for gradient estimation.

Detecting Ties in k-Nearest Neighbor Search

`knnsearch` now optionally returns all kth nearest neighbors of points, instead of just one. The `knnsearch` methods for `ExhaustiveSearcher` and `KDTreeSearcher` also have this option.

Distribution Fitting Tool Uses fitdist Function

MATLAB functions generated with the Distribution Fitting Tool now use the `fitdist` function to create fitted probability distribution objects. The generated functions return probability distribution objects as output arguments.

Speed and Accuracy Improvements in Noncentral Chi-Square CDF

`ncx2cdf` is now faster and more accurate for large values of the noncentrality parameter.

Perfect Separation in Binomial Regression

If the two categories in a binomial regression model (such as `logit` or `probit`) are perfectly separated, the best-fitting model is degenerate with infinite coefficients. In this case, the `glmfit` function is likely to exceed its iteration limit. `glmfit` now tries to detect this perfect separation and display a diagnostic message.

Sign Convention in mdscale

`mdscale` now enforces that, in each column of the output `Y`, the value with the largest magnitude has a positive sign. This change makes results consistent across releases and platforms—small changes used to lead to sign reversals.

Demo of Credit Rating Classification Via Bagged Decision Trees

The credit rating demo that used to be exclusively in Financial Toolbox™ is now available in Statistics Toolbox. The demo uses bagged decision trees for classifying creditworthiness.

To view the demo at the MATLAB command line, enter:

```
showdemo creditratingdemo
```

R2010b

Version: 7.4

New Features: Yes

Bug Fixes: Yes

Parallel Computing Support for More Functions

Statistics Toolbox now supports parallel execution for the following functions:

- `candexch`
- `cordexch`
- `daugment`
- `dcovary`
- `nnmf`
- `plsregress`
- `rowexch`
- `sequentialfs`

For more information, see the Parallel Statistics chapter in the User's Guide.

Algorithm to Rank Features in Classification and Regression

New filter algorithm, `relieff`, is based on nearest neighbors. The ReliefF algorithm accounts for correlations among predictors by computing the effect of every predictor on the class label (or true response for regression) locally and then integrates these local estimates over the entire predictor space.

nlmefit Support for Error Models, and nlmefitsa changes

Compatibility Considerations: Yes

`nlmefit` now supports the following error models:

- `combined`
- `constant`
- `exponential`
- `proportional`

You can specify an error model with both `nlmefitsa` and `nlmefit`.

The `nlmefit` `bic` calculation has changed. Now the degrees of freedom value is based on the number of groups rather than the number of observations. This conforms with the `bic` definition used by the `nlmefitsa` function.

Both `nlmefit` and `nlmefitsa` now store the estimated error parameters in the `errorparm` field of the output `stats` structure. The `rmse` field of the structure now contains the root mean squared residual for all error models; this value is computed on the log scale for the `exponential` model.

Compatibility Considerations

In the previous release, the `rmse` field was used by `nlmefitsa` for both mean squared residual and the estimated error parameter. Change your code, if necessary, to address the appropriate field in the `stats` structure.

As described in “nlmefit Support for Error Models, and nlmefitsa changes” on page 44, `nlmefit` now calculates different `bic` values than in previous releases.

Surrogate Splits for Decision Trees

The new surrogate splits feature in `classregtree` allows for better handling of missing values, more accurate estimation of variable importance, and calculation of the predictive measure of association between variables.

New Bagged Decision Tree Properties

`TreeBagger` and `CompactTreeBagger` classes have two new properties:

- `NVarSplit` provides the number of decision splits for each predictor variable.
- `VarAssoc` provides a measure of association between pairs of predictor variables.

Enhanced Cluster Analysis Performance

The `linkage` function has improved performance for the `centroid`, `median`, and `single` linkage methods.

The `linkage` and `pdist` hierarchical cluster analysis functions support larger array dimensions with 64-bit platforms, so can handle larger problems.

Export Probability Objects with dfittool

Compatibility Considerations: Yes

The distribution fitting GUI (`dfittool`) now allows you to export fits to the MATLAB workspace as probability distribution fit objects. For more information, see Modeling Data Using the Distribution Fitting Tool.

Compatibility Considerations

If you load a distribution fitting session that was created with previous versions of Statistics Toolbox, you cannot save an existing fit. Fit the distribution again to enable saving.

Compute Partial Correlation of Two Variables Correcting for All Other Variables

`partialcorr` now accepts a new syntax, `RHO = partialcorr(X)`, which returns the sample linear partial correlation coefficients between pairs of variables in `X`, controlling for the remaining variables in `X`. For more information, see the function reference page.

Specify Number of Evenly Spaced Quantiles

`quantile` now accepts a new syntax, `Y = quantile(X,N,...)`, which returns quantiles at the cumulative probabilities (1:N)/(N+1) where `N` is a scalar positive integer value.

Control Location and Orientation of Marginal Histograms with scatterhist

`scatterhist` now accepts three parameter name/value pairs that control where and how the histogram plots appear. The new parameter names are `NBins`, `Location`, and `Direction`. For more information, see the function reference page.

Return Bootstrapped Statistics with bootci

`bootci` has a new output option which returns the bootstrapped statistic computed for each of the `NBoot` bootstrap replicate samples. For more information, see the function reference page.

R2010a

Version: 7.3

New Features: Yes

Bug Fixes: Yes

Stochastic Algorithm Functionality in NLME Models

New stochastic algorithm for fitting NLME models is more robust with respect to starting values, enables parameter transformations, and relaxes assumption of constant error variance. See `nlmefitsa`.

k-Nearest Neighbor Searching

New functions for *k*-Nearest Neighbor (*k*NN) search efficiently to find the closest points to any query point. For information, see k-Nearest Neighbor Search and Radius Search.

Confidence Intervals Option in perfcurve

A new option in the `perfcurve` function computes confidence intervals for classifier performance curves.

Observation Weights Options in Resampling Functions

New options to weight resampling probabilities broaden the range of models supported by `bootstrp`, `bootci`, and `perfcurve` functions.

R2009b

Version: 7.2

New Features: Yes

Bug Fixes: Yes

New Parallel Computing Support for Certain Functions

Statistics Toolbox now supports parallel execution for the following functions:

- `bootci`
- `bootstrp`
- `crossval`
- `jackknife`
- `TreeBagger`

For more information on parallel computing in the Statistics Toolbox, see Parallel Computing Support for Resampling Methods.

New Stack and Unstack Methods for Dataset Arrays

`dataset.unstack` converts a “tall” dataset array to an equivalent dataset array that is in "wide format", by "unstacking" a single variable in the tall dataset array into multiple variables in wide. `dataset.stack` reverses this manipulation by converting a “wide” dataset array to an equivalent dataset array that is in "tall format", by "stacking up" multiple variables in the wide dataset array into a single variable in tall.

New Support for SAS Transport (.xpt) Files

Statistics Toolbox now supports importing and exporting files in SAS Transport (.xpt) format. For more information, see the `xptread` and `dataset.export` reference pages.

New Output Function in nlmefit for Monitoring or Canceling Calculations

The `nlmefit` function now supports using an output function to monitor or cancel calculations. For more information, see the `nlmefit` reference page.

R2009a

Version: 7.1

New Features: Yes

Bug Fixes: Yes

Enhanced Dataset Functionality

- An enhanced `dataset.join` method provides additional types of join operations:
  - `join` can now perform more complicated inner and outer join operations that allow a many-to-many correspondence between dataset arrays `A` and `B`, and allow unmatched observations in either `A` or `B`.
  - `join` can be of `Type` `'inner'`, `'leftouter'`, `'rightouter'`, `'fullouter'`, or `'outer'` (which is a synonym for `'fullouter'`). For an inner join, the dataset array, `C`, only contains observations corresponding to a combination of key values that occurred in both `A` and `B`. For a left (or right) outer join, `C` also contains observations corresponding to keys in `A` (or `B`) that did not match any in `B` (or `A`).
  - `join` can now return index vectors indicating the correspondence between observations in `C` and those in `A` and `B`.
  - `join` now supports using multiple keys.
  - `join` now supports an optional parameter for specifying missing key behavior rather than raising an error.
- An enhanced `dataset.export` method now supports exporting directly to Microsoft® Excel® files.

New Naïve Bayes Classification

- The `NaiveBayes` classification object is suitable for data sets that contain many predictors or features.
- It supports normal, kernel, multinomial, and multivariate multinomial distributions.

New Ensemble Methods for Classification and Regression Trees

- New classification objects, `TreeBagger` and `CompactTreeBagger`, provide improved performance through bootstrap aggregation (bagging).
- Includes Breiman's “random forest” method.
- Enhanced `classregtree` has more options for growing and pruning trees.

New Performance Curve Function

- New `perfcurve` function provides graphical method to evaluate classification results.
- Includes ROC (receiver operating characteristic) and other curves.

New Probability Distribution Objects

- Provides a consistent interface for working with probability distributions.
- Can be created directly using the `ProbDistUnivParam` constructor, or fit to data using the `fitdist` function.
- Option to fit distributions by group.
- Includes kernel object methods and parametric object methods that you can use to analyze the distribution represented by the object.
- Includes kernel object properties and parametric object properties that you can access to determine the fit results and evaluate their accuracy.
- Related enhancements in the `chi2gof`, `histfit`, `kstest`, `probplot`, and `qqplot` functions.

R2008b

Version: 7.0

New Features: Yes

Bug Fixes: No

Classification

The new `confusionmat` function tabulates misclassifications by comparing known and predicted classes of observations.

Data Organization

Compatibility Considerations: Yes

Dataset arrays constructed by the `dataset` function can now be written to an external text file using the new `export` function.

When reading external text files into a dataset array, `dataset` has a new `'TreatAsEmpty'` parameter for specifying strings to be treated as empty.

Compatibility Considerations

In previous versions, `dataset` used `eval` to evaluate strings in external text files before writing them into a dataset array. As a result, strings such as `'1/1/2008'` were treated as numerical expressions with two divides. Now, `dataset` treats such expressions as strings, and writes a string variable into the dataset array whenever a column in the external file contains a string that does not represent a valid scalar value.

Model Assessment

The cross-validation function, `crossval`, has new options for directly specifying loss functions for mean-squared error or misclassification rate, without having to provide a separate function M-file.

Multivariate Methods

The `procrustes` function has new options for computing linear transformations without scale or reflection components.

Probability Distributions

Compatibility Considerations: Yes

The multivariate normal functions mvnpdf, mvncdf, and mvnrnd now accept vector specification of diagonal covariance matrices, with corresponding gains in computational efficiency.

The hypergeometric distribution has been added to both the disttool and randtool graphical user interfaces.

Compatibility Considerations

The ksdensity function may give different answers for the case where there are censoring times beyond the last observed value. In this case, ksdensity tries to reduce the bias in its density estimate by folding kernel functions across a folding point so that they do not extend into the area that is completely censored. Two things have changed for this release:

1. In previous releases the folding point was the last observed value. In this release it is the first censoring time after the last observed value.

2. The folding procedure is applied not just when the 'function' parameter is 'pdf', but for all 'function' values.

Regression Analysis

The new nlmefit function fits nonlinear mixed-effects models to data with both fixed and random sources of variation. Mixed-effects models are commonly used with data over multiple groups, where measurements are correlated within groups but independent between groups.

Statistical Visualization

Compatibility Considerations: Yes

The boxplot function has new options for handling multiple grouping variables and extreme outliers.

The lsline, gline, refline, and refcurve functions now work with scatter plots produced by the scatter function. In previous versions, these functions worked only with scatter plots produced by the plot function.

The following visualization functions now have custom data cursors, displaying information such as observation numbers, group numbers, and the values of related variables:

- `andrewsplot`
- `biplot`
- `ecdf`
- `glyphplot`
- `gplotmatrix`
- `gscatter`
- `normplot`
- `parallelcoords`
- `probplot`
- `qqplot`
- `scatterhist`
- `wblplot`

Compatibility Considerations

Changes to `boxplot` have altered a number of default behaviors:

- Box labels are now drawn as text objects rather than tick labels. Any code that customizes the box labels by changing tick marks should now set the tick locations as well as the tick labels.
- The function no longer returns a handles array with a fixed number handles, and the order and meaning of the handles now depends on which options are selected. To locate a handle of interest, search for its `'Tag'` property using `findobj`. `'Tag'` values for box plot components are listed on the `boxplot` reference page.
- There are now valid handles for outliers, even when boxes have no outliers. In previous releases, the handles array returned by the function had `NaN` values in place of handles when boxes had no outliers. Now the `'xdata'` and `'ydata'` for outliers are `NaN` when there are no outliers.

- For small groups, the `'notch'` parameter sometimes produces notches that extend outside of the box. In previous releases, the notch was truncated to the extent of the box, which could produce a misleading display. A new value of `'markers'` for this parameter avoids the display issue.

As a consequence, the `anova1` function, which displays notched box plots for grouped data, may show notches that extend outside the boxes.

Utility Functions

The statistics options structure created by `statset` now includes a `Jacobian` field to specify whether or not an objective function can return the Jacobian as a second output.

R2008a

Version: 6.2

New Features: Yes

Bug Fixes: Yes

Descriptive Statistics

Compatibility Considerations: Yes

Bootstrap confidence intervals computed by `bootci` are now more accurate for lumpy data.

Compatibility Considerations

The formula for `bootci` confidence intervals of type `'bca'` or `'cper'` involves the proportion of bootstrap statistics less than the observed statistic. The formula now takes into account cases where there are many bootstrap statistics exactly equal to the observed statistic.

Model Assessment

Two new cross-validation functions, `cvpartition` and `crossval`, partition data and assess models in regression, classification, and clustering applications.

Multivariate Methods

A new sequential feature selection function, `sequentialfs`, selects predictor subsets that optimize user-defined prediction criteria.

The new `nnmf` function performs nonnegative matrix factorization (NMF) for dimension reduction.

Probability Distributions

The new `sobolset` and `haltonset` functions produce quasi-random point sets for applications in Monte Carlo integration, space-filling experimental designs, and global optimization. Options allow you to skip, leap over, and scramble the points. The `qrandstream` function provides corresponding quasi-random number streams for intermittent sampling.

Regression Analysis

The new plsregress function performs partial least-squares regression for data with correlated predictors.

Statistical Visualization

The normspec function now shades regions of a normal density curve that are either inside or outside specification limits.

Utility Functions

The statistics options structure created by statset now includes fields for TolTypeFun and TolTypeX, to specify tolerances on objective functions and parameter values, respectively.

R2007b

Version: 6.1

New Features: Yes

Bug Fixes: Yes

Cluster Analysis

Compatibility Considerations: Yes

The new `gmdistribution` class represents Gaussian mixture distributions, where random points come from different multivariate normal distributions with certain probabilities. The `gmdistribution` constructor creates mixture models with specified means, covariances, and mixture proportions, or by fitting a mixture model with a specified number of components to data. Methods for the class include:

- `fit` — Distribution fitting function
- `pdf` — Probability density function
- `cdf` — Cumulative distribution function
- `random` — Random number generator
- `cluster` — Data clustering
- `posterior` — Cluster posterior probabilities
- `mahal` — Mahalanobis distance

The `cluster` function for hierarchical clustering now accepts a vector of cutoff values, and returns a matrix of cluster assignments, with one column per cutoff value.

Compatibility Considerations

The `kmeans` function now returns a vector of cluster indices of length *n*, where *n* is the number of rows in the input data matrix `X`, even when `X` contains `NaN` values. In the past, rows of `X` with `NaN` values were ignored, and the vector of cluster indices was correspondingly reduced in size. Now the vector of cluster indices contains `NaN` values where rows have been ignored, consistent with other toolbox functions.

Design of Experiments

A new option in the *D*-optimal design function `candexch` specifies fixed design points in the row-exchange algorithm. A similar feature is already available for the `daugment` function, which uses the coordinate-exchange algorithm.

Hypothesis Tests

Compatibility Considerations: Yes

The kstest function now uses a more accurate method to calculate the *p*-value for a single-sample Kolmogorov-Smirnov test.

Compatibility Considerations

kstest now compares the computed *p*-value to the desired cutoff, rather than comparing the test statistic to a table of values. Results may differ from those in previous releases, especially for small samples in two-sided tests where an asymptotic formula was used in the past.

Probability Distributions

Compatibility Considerations: Yes

A new fitting function, copulafit, has been added to the family of functions that describe dependencies among variables using copulas. The function fits parametric copulas to data, providing a link between models of marginal distributions and models of data correlations.

A number of probability functions now have improved accuracy, especially for extreme parameter values. The functions are:

- betainv — More accurate for probabilities in P near 1.
- binocdf — More efficient and less likely to run out of memory for large values in X.
- binopdf — More accurate when the probabilities in P are on the order of eps.
- fcdf — More accurate when the parameter ratios V2./V1 are much less than the values in X.
- ncx2cdf — More accurate in some extreme cases that previously returned 0.
- poisscdf — More efficient and less likely to run out of memory for large values in X.

- `tcdf` — More accurate when the squares of the values in `X` are much less than the parameters in `V`.
- `tinv` — More accurate when the probabilities in `P` are very close to 0.5 and the outputs are very small in magnitude.

Function-style syntax for `paretotails` objects has been removed.

Compatibility Considerations

The changes to the probability functions listed above may lead to different, but more accurate, outputs than in previous releases.

In previous releases, syntax of the form `obj(x)` for a `paretotails` objects `obj` invoked the `cdf` method. This syntax now produces a warning. To evaluate the cumulative distribution function, use the syntax `cdf(obj,x)`.

Regression Analysis

Compatibility Considerations: Yes

The new `corrcov` function converts a covariance matrix to the corresponding correlation matrix.

The `mvregress` function now supports an option to force the estimated covariance matrix to be diagonal.

Compatibility Considerations

In previous releases the `mvregress` function, when using the `'cwls'` algorithm, estimated the covariance of coefficients `COVB` using the estimated, rather than the initial, covariance of the responses `SIGMA`. The initial `SIGMA` is now used, and `COVB` differs to a degree dependent on the difference between the initial and final estimates of `SIGMA`.

Statistical Visualization

The `boxplot` function has a new `'compact'` plot style suitable for displaying large numbers of groups.

R2007a

Version: 6.0

New Features: Yes

Bug Fixes: Yes

Data Organization

New categorical and dataset arrays are available for organizing and processing statistical data.

- Categorical arrays facilitate the use of nominal and ordinal categorical data.
- Dataset arrays provide a natural way to encapsulate heterogeneous statistical data and metadata, so that it can be accessed and manipulated using familiar methods analogous to those for numerical matrices.
- Categorical and dataset arrays are supported by a variety of new functions for manipulating the encapsulated data.
- Categorical arrays are now accepted as input arguments in all Statistics Toolbox functions that make use of grouping variables.

Hypothesis Testing

Expanded options are available for linear hypothesis testing.

- The new `linhyptest` function performs linear hypothesis tests on parameters such as regression coefficients. These tests have the form `H*b = c` for specified values of `H` and `c`, where `b` is a vector of unknown parameters.
- The `covb` output from `regstats` and the `SIGMA` output from `nlinfit` are suitable for use as the covariance matrix input argument required by `linhyptest`. The following functions have been modified to return a `covb` output for use with `linhyptest`: `coxphfit`, `glmfit`, `mnrfit`, `robustfit`.

Multivariate Statistics

Compatibility Considerations: Yes

The new `cholcov` function computes a Cholesky-like decomposition of a covariance matrix, even if the matrix is not positive definite. Factors are useful in many of the same ways as Cholesky factors, such as imposing correlation on random number generators.

The `classify` function for discriminant analysis has been improved.

- The function now computes the coefficients of the discriminant functions that define boundaries between classification regions.
- The output of the function is now of the same type as the input grouping variable `group`.

Compatibility Considerations

The `classify` function now returns outputs of different type than it did in the past. If the input argument `group` is a logical vector, output is now converted to a logical vector. In the past, output was returned as a cell array of `0`s and `1`s. If `group` is numeric, the output is now converted to the same type. For example, if `group` is of type `uint8`, the output will be of type `uint8`.

Probability Distributions

New `paretotails` objects are available for modeling distributions with an empirical cdf or similar distribution in the center and generalized Pareto distributions in the tails.

- The `paretotails` function converts a data sample to a `paretotails` object. The objects are useful for generating random samples from a distribution similar to the data, but with tail behavior that is less discrete than the empirical distribution.
- Objects from the `paretotails` class are supported by a variety of new methods for working with the piecewise distribution.
- The `paretotails` class provides function-like behavior, so that `p(x)` evaluates the cdf of `p` at values `x`.

Regression Analysis

Compatibility Considerations: Yes

The new `mvregresslike` function is a utility related to the `mvregress` function for fitting regression models to multivariate data with missing values. The new function computes the objective (log likelihood) function, and can also compute the estimated covariance matrix for the parameter estimates.

New classregtree objects are available for creating and analyzing classification and regression trees.

- The classregtree function fits a classification or regression tree to training data. The objects are useful for predicting response values from new predictors.
- Objects from the classregtree class are supported by a variety of new methods for accessing information about the tree.
- The classregtree class provides function-like behavior, so that t(X) evaluates the tree t at predictor values in X.
- The following functions now create or operate on objects from the new classregtree class: treefit, treedisp, treeval, treefit, treeprune, treetest.

Compatibility Considerations

Objects from the classregtree class are intended to be compatible with the structure arrays that were produced in previous versions by the classification and regression tree functions listed above. In particular, classregtree supports dot indexing of the form t.property to obtain properties of the object t. The class also provides function-like behavior through parenthesis indexing, so that t(x) uses the tree t to classify or compute fitted values for predictors x, rather than index into t as a structure array as it did in the past. As a result, cell arrays should now be used to aggregate classregtree objects.

Statistical Visualization

The new scatterhist function produces a scatterplot of 2D data and illustrates the marginal distributions of the variables by drawing histograms along the two axes. The function is also useful for viewing properties of random samples produced by functions such as copularnd, mvnrnd, and lhsdesign.

Other Improvements

- The `mvtrnd` function now produces a single random sample from the multivariate *t* distribution if the `cases` input argument is absent.
- The `zscore` function, which centers and scales input data by mean and standard deviation, now returns the means and standard deviations as additional outputs.

R2006b

Version: 5.3

New Features: Yes

Bug Fixes: Yes

Demos

The following demo has been updated:

- Selecting a Sample Size — Modified to highlight the new `sampsizepwr` function

Design of Experiments

The following visualization functions, commonly used in the design of experiments, have been added:

- `interactionplot` — Two-factor interaction plot for the mean
- `maineffectsplot` — Main effects plot for the mean
- `multivarichart` — Multivari chart for the mean

Hypothesis Tests

Compatibility Considerations: Yes

The following functions for hypothesis testing have been added or improved:

- `jbtest` — Replaces the chi-square approximation of the test statistic, which is asymptotic, with a more accurate algorithm that interpolates *p*-values from a table of quantiles. A new option allows you to run Monte Carlo simulations to compute *p*-values outside of the table.
- `lillietest` — Uses an improved version of Lilliefors' table of quantiles, covering a wider range of sample sizes and significance levels, with more accurate values. New options allow you to test for exponential and extreme value distributions, as well as normal distributions, and to run Monte Carlo simulations to compute *p*-values outside of the tables.
- `runstest` — Adds a test for runs up and down to the existing test for runs above or below a specified value.
- `sampsizepwr` — New function to compute the sample size necessary for a test to have a specified power. Options are available for choosing a variety of test types.

Compatibility Considerations

If the significance level for a test lies outside the range of tabulated values, [0.001, 0.5], then both `jbtest` and `lillietest` now return an error. In previous versions, `jbtest` returned an approximate *p*-value and `lillietest` returned an error outside a smaller range, [0.01, 0.2]. Error messages suggest using the new Monte Carlo option for computing values outside the range of tabulated values.

If the data sample for a test leads to a *p*-value outside the range of tabulated values, then both `jbtest` and `lillietest` now return, with a warning, either the smallest or largest tabulated value. In previous versions, `jbtest` returned an approximate *p*-value and `lillietest` returned `NaN`.

Multinomial Distribution

The multinomial distribution has been added to the list of almost 50 probability distributions supported by the toolbox.

- `mnpdf` — Multinomial probability density function
- `mnrnd` — Multinomial random number generator

Regression Analysis

Multinomial Regression

Support has been added for multinomial regression modeling of discrete multi-category response data, including multinomial logistic regression. The following new functions supplement the regression models in `glmfit` and `glmval` by providing for a wider range of response values:

- `mnrfit` — Fits a multinomial regression model to data
- `mnrval` — Computes predicted probabilities for the multinomial regression model

Multivariate Regression

The new mvregress function carries out multivariate regression on data with missing response values. An option allows you to specify how missing data is handled.

Survival Analysis

coxphfit — A new option allows you to specify the values at which the baseline hazard is computed.

Statistical Process Control

Compatibility Considerations: Yes

The following new functions consolidate and expand upon existing functions for statistical process control:

- capability — Computes a wider range of probabilities and capability indices than the capable function found in previous releases
- controlchart — Displays a wider range of control charts than the ewmaplot, schart, and xbarplot functions found in previous releases
- controlrules — Supplements the new controlchart function by providing for a wider range of control rules (Western Electric and Nelson)
- gagerr — Performs a gage repeatability and reproducibility study on measurements grouped by operator and part

Compatibility Considerations

The capability function subsumes the capable function that appeared in previous versions of Statistics Toolbox software, and the controlchart function subsumes the functions ewmaplot, schart, and xbarplot. The older functions remain in the toolbox for backwards compatibility, but they are no longer documented or supported.

R2006a

Version: 5.2

New Features: Yes

Bug Fixes: Yes

Analysis of Variance

Support for nested and continuous factors has been added to the `anovan` function for *N*-way analysis of variance.

Bootstrapping

The following functions have been added to supplement the existing `bootstrp` function for bootstrap estimation:

- `bootci` — Computes confidence intervals of a bootstrapped statistic. An option allows you to choose the type of the bootstrap confidence interval.
- `jackknife` — Draws jackknife samples from a data set and computes statistics on each sample

Demos

The following demos have been added to the toolbox:

- Bayesian Analysis for a Logistic Regression Model
- Time Series Regression of Airline Passenger Data

The following demo has been updated to demonstrate new features:

- Random Number Generation

Design of Experiments

The new `fracfactgen` function finds a set of fractional factorial design generators suitable for fitting a specified model.

The following functions for *D*-optimal designs have been enhanced:

- `cordexch`, `daugment`, `dcovary`, `rowexch` — New options specify the range of values and the number of levels for each factor, exclude factor combinations, treat factors as categorical rather than continuous, control the number of iterations, and repeat the design generation process from random starting points

- candexch — New options control the number of iterations and repeat the design generation process from random starting points
- candgen — New options specify the range of values and the number of levels for each factor, and treat factors as categorical rather than continuous
- x2fx — New option treats factors as categorical rather than continuous

Hypothesis Tests

The new dwtest function performs a Durbin-Watson test for autocorrelation in linear regression.

Multivariate Distributions

Two new functions have been added to compute multivariate cdfs. These supplement existing functions for pdfs and random number generators for the same distributions.

- mvncdf — Cumulative distribution function for the multivariate normal distribution
- mvtcdf — Cumulative distribution function for the multivariate *t* distribution

Random Number Generation

Copulas

New functions have been added to the toolbox that allow you to use copulas to model correlated multivariate data and generate random numbers from multivariate distributions.

- copulacdf — Cumulative distribution function for a copula
- copulaparam — Copula parameters as a function of rank correlation
- copulapdf — Probability density function for a copula
- copularnd — Random numbers from a copula
- copulastat — Rank correlation for a copula

Markov Chain Monte Carlo Methods

The following functions generate random numbers from nonstandard distributions using Markov Chain Monte Carlo methods:

- `mhsample` — Generate random numbers using the Metropolis-Hasting algorithm
- `slicesample` — Generate random numbers using a slice sampling algorithm

Pearson and Johnson Systems of Distributions

Support has been added for random number generation from Pearson and Johnson systems of distributions.

- `pearsrnd` — Random numbers from a distribution in the Pearson system
- `johnsrnd` — Random numbers from a distribution in the Johnson system

Robust Regression

To supplement the `robustfit` function, the following functions now have options for robust fitting:

- `nlinfit` — Nonlinear least-squares regression
- `nlparci` — Confidence intervals for parameters in nonlinear regression
- `nlpredci` — Confidence intervals for predictions in nonlinear regression

Statistical Process Control

The following control chart functions now support time-series objects:

- `xbarplot` — Xbar plot
- `schart` — Standard deviation chart
- `ewmaplot` — Exponentially weighted moving average plot

R14SP3

Version: 5.1

New Features: Yes

Bug Fixes: No

Demos

The following demos have been added to the toolbox:

- Curve Fitting and Distribution Fitting
- Fitting a Univariate Distribution Using Cumulative Probabilities
- Fitting an Orthogonal Regression Using Principal Components Analysis
- Modelling Tail Data with the Generalized Pareto Distribution
- Pitfalls in Fitting Nonlinear Models by Transforming to Linearity
- Weighted Nonlinear Regression

The following demo has been updated:

- Modelling Data with the Generalized Extreme Value Distribution

Descriptive Statistics

The new `partialcorr` function computes the correlation of one set of variables while controlling for a second set of variables.

The `grpstats` function now computes a wider variety of descriptive statistics for grouped data. Choices include the mean, standard error of the mean, number of elements, group name, standard deviation, variance, confidence interval for the mean, and confidence interval for new observations. The function also supports the computation of user-defined statistics.

Hypothesis Tests

Chi-Square Goodness-of-Fit Test

The new `chi2gof` function tests if a sample comes from a specified distribution, against the alternative that it does not come from that distribution, using a chi-square test statistic.

Variance Tests

Three functions have been added to test sample variances:

- `vartest` — One-sample chi-square variance test. Tests if a sample comes from a normal distribution with specified variance, against the alternative that it comes from a normal distribution with a different variance.
- `vartest2` — Two-sample *F*-test for equal variances. Tests if two independent samples come from normal distributions with the same variance, against the alternative that they come from normal distributions with different variances.
- `vartestn` — Bartlett multiple-sample test for equal variances. Tests if multiple samples come from normal distributions with the same variance, against the alternative that they come from normal distributions with different variances.

Ansari-Bradley Test

The new `ansaribradley` function tests if two independent samples come from the same distribution, against the alternative that they come from distributions that have the same median and shape but different variances.

Tests of Randomness

The new `runstest` function tests if a sequence of values comes in random order, against the alternative that the ordering is not random.

Probability Distributions

Support has been added for two new distributions:

- "Generalized Extreme Value Distribution" on page 87
- "Generalized Pareto Distribution" on page 88

Generalized Extreme Value Distribution

The Generalized Extreme Value distribution combines the Gumbel, Frechet, and Weibull distributions into a single distribution. It is used to model extreme values in data.

The following distribution functions have been added:

- `gevcdf` — Cumulative distribution function
- `gevfit` — Parameter estimation function
- `gevinv` — Inverse cumulative distribution function
- `gevlike` — Negative log-likelihood function
- `gevpdf` — Probability density function
- `gevrnd` — Random number generator
- `gevstat` — Distribution statistics

Generalized Pareto Distribution

The Generalized Pareto distribution is used to model the tails of a data distribution.

The following distribution functions have been added:

- `gpcdf` — Cumulative distribution function
- `gpfit` — Parameter estimation function
- `gpinv` — Inverse cumulative distribution function
- `gplike` — Negative log-likelihood function
- `gppdf` — Probability density function
- `gprnd` — Random number generator
- `gpstat` — Distribution statistics

Regression Analysis

- The new `coxphfit` function fits Cox's proportional hazards regression model to data.
- The new `invpred` function estimates the inverse prediction intervals for simple linear regression.
- The `polyconf` function has new options to let you specify the confidence interval computed.

Statistical Visualization

Both the `ecdf` and `ksdensity` functions now produce plots when no output arguments are specified.

R14SP2

Version: 5.0.2

New Features: Yes

Bug Fixes: Yes

Multivariate Statistics

The `cophenet` function now returns cophenetic distances as well as the cophenetic correlation coefficient.